REMARKS

This Amendment is in response to the non-final Office action (Paper No. 20080309) mailed on 24 October 2008. Reexamination and reconsideration are respectfully requested.

Listing of The Claims

Pursuant to 37 CFR §121(c), the claim listing, including the text of the claims, will serve to replace all prior versions of the claims, in the application.

Status of The Claims

Claims 1-10 are pending in this application.

Amendment of The Claims

Claims 1-10 are amended.

Issues Raised by Paper No. 20080309

I. Claim Rejections - 35 USC § 102

Claim 1 is rejected under 35 U.S.C. 102 (e) as being anticipated by Civanlar et al. (US Patent No. 6,078,963).

I-1. Brief Summary of the Invention

The present invention relates to a distributed router and a process that dynamically manages forwarding information, with each routing node sharing its collected routing information in real time with the other routing nodes and managing forwarding information dynamically based on the routing information, thereby avoiding a need for packet forwarding in order to share the routing information between the routing nodes. The routing information

is selectively updated in forwarding tables; thus efficiently managing the forwarding table of each routing node. Furthermore, the size of the forwarding table in each routing node may be reduced because the forwarding information of each routing node is managed in the form of a binary aggregation tree and the aggregation level of a delegation node that aggregates node information corresponding to routing information in the aggregation tree, may be variably set. In other words, the present invention is directed to sorting nodes inserted into a routing table according to their ancestor nodes and inserting the sorted node into a forwarding table.

I-2. Brief Summary of the Cited Prior Art

The primary reference cited by the Examiner is Civanlar '963. Civanlar '963 relates to a network router having a plurality of intelligent router ports. Each intelligent router port may have its own routing and/or forwarding engines. Thus, a centralized master routing and forwarding engine, existing in conventional routers, is not necessary. Accordingly, the bottlenecking problems associated with a centralized routing and forwarding engine are significantly reduced.

The secondary reference cited by the Examiner is Venkatachary '184. Venkatachary '184 relates to methods and devices for layer four switching in a router. In a first method, a grid of tries, which are binary branching trees, is constructed from the set of routing filters. The grid includes a dest-trie and a number of source tries. To avoid memory blowup, each filter is stored in exactly one trie. The tries are traversed to find the lowest cost routing.

Switch pointers are used to improve the search cost. In an extension of this method, hash tables may be constructed that point to grid-of-tries structures. The hash tables may be used to handle combinations of port fields and protocol fields. Another method is based on hashing, in which searches for lowest cost matching filters take place in bit length tuple space. Rectangle searching with precomputation and markers are used to eliminate a whole column of tuple space when a match occurs, and to eliminate the rest of a row when no match is found. Various optimizations of these methods are also provided. A router incorporating memory and processors implementing these methods is capable of rapid, selective switching of data packets on various types of networks, and is particularly suited to switching on Internet Protocol networks.

I-3. Summary of Differences Between Invention and Cited Prior Art

1. One of the principal patentable distinctions between the pending claims and the combined prior art is that the pending claims' "switching module" is shared by a plurality of routing nodes, while Civanlar '963's "forward engine", which is alleged by the Examiner to be Applicant's "switching module", is not shared by a plurality of router ports. More specifically, the switching module of the pending claims has a plurality of routing protocol processing units that correspond to each of a plurality of routing nodes because the switching module is shared by a plurality of routing nodes. See the pending claims:

> ***claim 1***, "a switching module having **a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol**

> **processing units of each of the routing nodes**"; and
>
> ***claim 9***, "with each routing node having **a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units in said switching module**".

On the other hand, a forward engine 105 of Civanlar '963 is included into its own router port 103 such that the forward engine itself has its own routing protocol only. In other words, the forward engine of Civanlar '963 does not have a routing protocol of another forward engine 105.

In short, the pending claims' switching module is shared by a plurality of nodes but the forward engine of Civanlar '963 is not shared by a plurality of router ports. Therefore, Civanlar '963's "forward engine" is not synonymous with Applicant's "switching module".

2. Another principal patentable distinction between the pending claims and the combined prior art is that in the pending claims each routing node shares routing information collected by itself with the switching module and with the other routing nodes. On the other hand, the combined prior art does not teach or suggest that the router ports can share routing information with a central switching module. See the pending claims:

> ***claim 1***, "a plurality of routing nodes each having a plurality of routing protocol processing units, with each routing protocol processing unit processing data in accordance with a respectively corresponding routing protocol; and
>
> a switching module having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units of each of the routing nodes, with the switching module disposed to share in

real time routing information collected by each of the routing nodes with others of the routing nodes";

claim 9, "a switching module accommodating a plurality of routing protocol processing units while managing forwarding information within the distributed architecture router, with each routing protocol processing unit processing data in accordance with a respectively corresponding routing protocol; and

a plurality of routing nodes each disposed to service networks within corresponding source areas comprised of local areas , with each routing node having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units in said switching module to form a source area comprising a virtual area and share in real time collected routing information assembled by a routing table and an aggregation tree derived from said routing table".

3. Still another principal patentable distinction between the pending claims and the combined prior art is that the combined prior art does not teach or suggest the pending claims' "*inserting new routing information in a routing table*". Venkatachary '184, which is cited by the Examiner in this regard, merely teaches placing a switch pointer (i.e., a binary flag) in a prefix trie, and the switch pointers are used for allowing direct jump from a first node to a second node when the first node does not match a prefix string. The Examiner's attention is invited to note that "routing information" is an art recognized term which is used by those skill in the art, and which is defined as "*the information needed to define a network route*". On the contrary, it is not clear whether or not Venkatachary '184's "switch pointer" does not define a network route. In addition, a brief searching of "routing information" in USPTO Patent Database reveals more than 200 U.S. Patents or Patent Publications that use

the term "routing information". For example, U.S. Patent No. 7,382,769, which is entitled Automatic filtering to prevent network attacks and issued on June 3, 2008, recited "the routing information defines network routes". Apparently, Applicant's "routing information" is not as simple as Venkatachary '184's "binary flags". In other words, Venkatachary '184's "placing switching pointers" is not synonymous with Applicant's "*inserting new routing information in a routing table*". See the pending claims:

> ***claim 2***, "when new routing information is to be inserted into a routing table in a distributed router"; and

> ***claim 10***, "said routing nodes responding to insertion of new routing information into said routing table".

4. A further principal patentable distinction between the pending claims and the combined prior art is that the combined prior art does not teach or suggest the pending claims' "*deleting routing information from a routing table*". Venkatachary '184, which is cited by the Examiner in this regard, merely mentions updating a least cost matching filter. It is not clear from Venkatachary '184 whether "updating a least cost matching filter" including a step of "deleting routing information from a routing table". See the pending claim:

> *claim 7*, "when routing information is to be deleted from a routing table".

I-4. Claim 1

Regarding claim 1, on pages 3-4 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 1, Civanlar discloses a distributed router comprising:
>
> a) a plurality of routing nodes each having a plurality of routing protocols; (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in a overall network (plurality of routers); col 3, 1137-41: any known types of routing protocols packets may be received (OSPF, RIP, BGP4)) and
>
> b) a switching module having a plurality of routing protocols corresponding to the routing protocols of each of the routing nodes, disposed to share in real time routing information collected by each of the routing nodes with others of the routing nodes. (Civanlar col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database; col 3, II 37-41: any known types of routing protocols packets may be received (OSPF, RIP,BGP4))".

Applicant respectfully traverses.

The pending claim 1 calls for, in part,

> "a plurality of routing nodes each having a plurality of routing protocol processing units, with each routing protocol processing unit processing data in accordance with a respective corresponding routing protocol; and
>
> a switching module having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units of each of the routing nodes, with the switching module disposed to share in real time routing information collected by each of the routing nodes with others of the routing nodes."

According to the pending claim 1 in connection with Applicant's original FIG. 3, a distributed router includes a plurality of routing nodes 210 and a switching module 250. Each one of a plurality of routing nodes 210 has a plurality of routing protocol processing units 211-214, and each routing protocol processing unit 211-214 processes data in

accordance with a respective corresponding routing protocol (i.e., RIPD, SDPFD, BGPD, or DGLUED). In addition, switching module 250 includes a plurality of routing protocol processing units 251-254 that are communicatively connected with routing protocol processing units 211-214 of each of the routing nodes, to share routing information collected by each of the routing nodes with other routing nodes.

First, respectfully, the prior art fails to disclose claim 1's "switching module". The pending claim's "switching module" is **shared by a plurality of routing nodes**, while Civanlar '963's "forward engine", which is alleged by the Examiner to be Applicant's "switching module", is **not shared by a plurality of router ports**. More specifically, the switching module of the pending claim has a plurality of routing protocol processing units that correspond to each of a plurality of routing nodes because the switching module is shared by a plurality of routing nodes. On the other hand, a forward engine 105 of Civanlar '963 is included into its own router port 103 such that the forward engine itself has its own routing protocol only. In other words, the forward engine of Civanlar '963 does not have a routing protocol of another forward engine 105.

In short, the pending claims' switching module is shared by a plurality of nodes but the forward engine of Civanlar '963 is not shared by a plurality of router ports. Therefore, Civanlar '963's "forward engine" is not synonymous with Applicant's "switching module"

Secondly, respectfully, Civanlar '963 fails to disclose the pending claim's "plurality of routing protocol processing units" and "each routing protocol processing unit processing data in accordance with a respective corresponding routing protocol". Civanlar '963's

column 3, lines 37-41 as cited by the Examiner in Paper No. 20080309 merely mentions different types of routing protocols packets.[1] Civanlar '963 does not disclose however, how to process those different types of routing protocols packets, or whether those different types of routing protocols packets are processed in the same processing unit or in different processing units. On the other hand, Applicant's claim 1 defines that each routing protocol processing unit processes data in accordance with a respectively corresponding routing protocol (i.e., RIPD, SDPFD, BGPD, or DGLUED). In other words, in the pending claim 1's distributed router, data with different protocol is processed in a corresponding routing protocol processing unit.

Thirdly, respectfully, Civanlar '963 fails to disclose the pending claim's "a plurality of routing protocol processing units in a switching module that are communicatively connected with the corresponding routing protocol processing units of each of the routing nodes". Civanlar '963 merely mentions that the switching fabric may partially or fully interconnect some or all of the router ports.[2] Civanlar '963 does not disclose however, whether there are a plurality of routing protocol processing units in the switching fabric, and how to interconnect the switching fabric with the router ports. On the other hand, Applicant's pending claim 1 clearly defines that the routing protocol processing units in the

[1] Civanlar '963's column 3, lines 37-41 reads: "Any known types of routing protocols packets may be received by the routing engine 107, such as those conforming to the routing Internet protocol (RIP), the open shortest path forwarding (OSPF) protocol, or the border gateway protocol 4 (BGP4)."

[2] Civanlar '963's column 3, lines 5-7 reads: "The switching fabric 102 may switch between and/or otherwise partially or fully interconnect some or all of the intelligent router ports 103."

switching module are connected with the corresponding routing protocol processing units in each of the routing nodes.

Fourthly, Civanlar '963 fails to disclose the pending claim's "switching module disposed to share routing information collected by each of the routing nodes with other routing nodes". Civanlar '963 merely discloses that the switching fabric may transfer data at a high speed.[3] In addition, Civanlar '963 explicitly discloses that the switching fabric does not have a central processor, and each router port may maintain its own routing tables without the need for a central processor coordinating this activity.[4] Civanlar '963 does not disclose that the switching fabric may share routing information of each router port with other router ports.

In summary, Civanlar '963 fails to disclose the pending claim's "plurality of routing protocol processing units", or "each routing protocol processing unit processing data in accordance with a respective corresponding routing protocol", or "a plurality of routing protocol processing units in a switching module that are communicatively connected with the corresponding routing protocol processing units of each of the routing nodes", or "the switching module disposed to share in real time routing information collected by each of the

---

[3] Civanlar '963's column 3, lines 7-8 reads: "The switching fabric 102 preferably transfers data at a very high speed."

[4] Civanlar '963's column 3, lines 19-27 reads: "Accordingly, embodiments of the present invention do not require the switching fabric 102 nor the router 100 itself to have a central processor as is required in conventional routers. Instead, as will be discussed in more detail below, each intelligent router port 103 may operate autonomously and may be capable of generating and maintaining its own routing tables and/or forwarding data packets in accordance with the routing tables, without the need for a central processor coordinating this activity."

routing nodes with others of the routing nodes".

As a result, the rejection of claim 1 should be withdrawn.

II. Claim Rejections- 35 USC § 103

Claims 2 - 10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Civanlar in view of Venkatachary et al. (US Patent No. 6,212,184).

II-1. Claim 2

Regarding claim 2, on pages 4-5 of Paper No. 20080309, the Examiner stated:

> "Civanlar does not explicitly disclose an aggregation tree for routing information.
>
> However, Venkatachary discloses:
>
> (1) wherein an aggregation tree based on the routing table, detecting a position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree; (Venkatachary col. 15, lines 50-60: search and update (insert) routing information)
>
> (2) determining presence and absence of an ancestor node of the insertion node at or below a predetermined maximum aggregation level; (Venkatachary col. 10, lines 6-33; col 16, 1126-36: determine existence ancestor node)
>
> (3) leaving the forwarding table un-updated with information about the insertion node in a presence of the ancestor node, when forwarding information is in the forwarding table and the insertion node and the ancestor node have been generated from a common source area; (Venkatachary col. 16, lines 37-52: switch pointer inserted when nil; no node exists)
>
> (4) in an absence of the ancestor node, resetting the aggregation level to a reset aggregation level not greater than the maximum aggregation level, and inserting a delegation node representative of the insertion node at the reset aggregation level; (Venkatachary col. 16, lines 26-36: switch pointer; reset to ancestor node level) and

(5) making an insertion of forwarding information by determining the source area of the inserted routing information, inserting forwarding information corresponding to the delegation node in the forwarding table when the source area of the routing information is a virtual area, and inserting forwarding information corresponding to the insertion node in the forwarding table when the source area of the routing information is a local area. (Venkatachary col. 15, lines 50-60: update (insert) forwarding information)".

Applicant respectfully traverses.

Claim 2 calls for, in part,

"(1) when new routing information is to be inserted into a routing table in a distributed router in which all routing nodes share a forwarding information made according to an aggregation tree based on the routing table, detecting a position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree;

(2) determining presence and absence of an ancestor node of the insertion node at or below a predetermined maximum aggregation level with respect to the insertion node;

(3) leaving the forwarding table un-updated with information about the insertion node in a presence of the ancestor node, when forwarding information corresponding to the ancestor node is in the forwarding table and both of the insertion node and the ancestor node have been generated from a common source area;

(4) in an absence of the ancestor node, resetting the aggregation level to a reset aggregation level not greater than the maximum aggregation level, and inserting forwarding information corresponding to a delegation node representative of the insertion node at the reset aggregation level in the forwarding table; and

(5) making an insertion of forwarding information by determining the source area of the routing information to be inserted, inserting forwarding information corresponding to the delegation node in the forwarding table when the source area of the routing information is a virtual area, and inserting forwarding information corresponding to the insertion node in

> the forwarding table when the source area of the routing information is a local area."

According to claim 2, when a piece of new routing information is to be inserted into a routing table, (1)first a position of an insertion node corresponding to the new routing information is detected; (2) it is then determined whether there is an ancestor node of the insertion node at or below a predetermined maximum aggregation level with respect to the insertion node; (3) if there is an ancestor node, the forwarding table is not updated with the information corresponding to the insertion node when both of the insertion node and the ancestor node have been generated from a common source area; (4) if there is no ancestor node, the aggregation level is reset and forwarding information corresponding to a delegation node representative of the insertion node is inserted at the reset aggregation level in the forwarding table; (5) the source area of the new routing information is determined, and when the source area of the new routing information is a virtual area, the forwarding information corresponding to the delegation node is inserted in the forwarding table, and when the source area of the new routing information is a local area, the forwarding information corresponding to the insertion node is inserted in the forwarding table.

First respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "inserting routing information in a routing table". In Paper No. 20080309, the Examiner explicitly admitted that Civanlar '963 fails to disclose an aggregation tree for routing information. On the other hand, Venkatachary '184 merely teaches how to place a switch pointer. In point of fact, Venkatachary '184's "switch pointer"

is different from the pending claim's "node". As explained in Venkatachary '184's specification, the switch pointer is pointed from a first node to a second node to enable a direct "jump" from the first node to the second node when the search on the first node fails.[5] There is no teaching in Venkatachary '184 regarding how to insert a piece of new routing information corresponding to a node.

Secondly, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "making an insertion of forwarding information by determining the source area of the routing information to be inserted". Venkatachary '184's column 15, lines 50-60 as cited by the Examiner merely mentions that the router may update the least cost matching filter.[6] A thorough search of Venkatachary '184's specification demonstrates that Venkatachary '184 does not disclose how to "update the least cost matching filter". Venkatachary '184 does not however, teach inserting new routing information, or determining whether the source area of the routing information is a virtual area or a local area.

In summary, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "inserting routing information in a routing table", or "making an insertion of forwarding information by determining the source area of the routing

---

[5] Venkatachary '184's column 16, lines 31-33 reads: "Intuitively, the switch pointers allow a jump directly to the lowest point in the ancestor source trie that has at least as good a source match as the current node."

[6] Venkatachary '184's column 15, lines 53-55 reads: "The router searches the source trie of D', and updates the least cost matching filter."

information to be inserted".

Consequently, the rejection of claim 2 is in error and should be withdrawn.

II-2. Claim 3

Regarding claim 3, on page 5 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 3, Civanlar discloses the method of claim 2. (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in an overall network; col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database) Civanlar does not explicitly disclose an aggregation tree for routing information. However, Venkatachary discloses wherein comprised of, after making said insertion of forwarding information when a delegation node is found to exist at the position of the insertion node while detecting a position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree, deleting from the forwarding table forwarding information corresponding to the delegation node. (Venkatachary col. 15, lines 50-60: update (insert, delete) forwarding information)".

Applicant respectfully traverses.

Claim 3 calls for, in part,

> "before making said insertion of forwarding information, and when a delegation node is found to exist at the position of the insertion node while detecting a position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree, deleting from the forwarding table forwarding information corresponding to the delegation node."

According to claim 3, before the forwarding information is inserted, and when a delegation node is found to exist at the position of the insertion node, the forwarding information

corresponding to the delegation node is deleted from the forwarding table.

Respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "deleting the forwarding information corresponding to the delegation node from the forwarding table before the forwarding information is inserted, and when a delegation node is found to exist at the position of the insertion node". As discussed previously, Venkatachary '184's column 15, lines 50-60 as cited by the Examiner merely mentions that the router may update the least cost matching filter,[7] and it is unclear from Venkatachary '184's specification whether "updating the least cost matching filter" is synonymous with "updating forwarding information". In addition, Venkatachary '184 does not teaches deleting forwarding information from the forwarding table.

Consequently, the rejection of claim 3 is in error and should be withdrawn.

II-3. Claim 4

Regarding claim 4, on page 6 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 4, Civanlar discloses the method of claim 2. (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in an overall network; col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database)
>
> Civanlar does not explicitly disclose an aggregation tree for routing information. However, Venkatachary discloses wherein comprised of: a) after making said insertion of forwarding information when a delegation node is found to exist

[7] Venkatachary '184's column 15, lines 53-55 reads: "The router searches the source trie of D', and updates the least cost matching filter."

> at the position of the insertion node while detecting said position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree, and when a left/right subtree of the delegation node exists, reinserting nodes of the left/right subtree, and deleting forwarding information corresponding to the delegation node from the forwarding table. (Venkatachary col. 10, lines 6-33; col 16, II 26-36: search forwarding information; col. 15, lines 50-60: update (insert, delete) forwarding information)".

Applicant respectfully traverses.

Claim 4 calls for, in part,

> "before making said insertion of forwarding information,. when a delegation node is found to exist at the position of the insertion node while detecting said position at which an insertion node corresponding to the new routing information is to be inserted into the aggregation tree, and when a left/right subtree of the delegation node exists,
> reinserting nodes of the left/right subtree, and
> deleting forwarding information corresponding to the delegation node from the forwarding table."

According to claim 4, before the forwarding information is inserted, and when a delegation node is found to exist at the position of the insertion node and a left/right subtree of the delegation node exists, the forwarding information corresponding to the delegation node is deleted from the forwarding table.

Respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "reinserting nodes of the left/right subtree, and deleting the forwarding information corresponding to the delegation node from the forwarding table before the forwarding information is inserted, and when a delegation node is found to exist at the position of the insertion node and a left/right subtree of the delegation node exists".

As discussed previously, Venkatachary '184's column 15, lines 50-60 as cited by the Examiner merely mentions that the router may update the least cost matching filter,[8] and it is unclear from Venkatachary '184's specification whether "updating the least cost matching filter" is synonymous with "updating forwarding information". In addition, Venkatachary '184 does not teaches deleting forwarding information from the forwarding table.

Consequently, the rejection of claim 4 is in error and should be withdrawn.

II-4. Claim 5

Regarding claim 5, on pages 7-8 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 5, Civanlar discloses the method of claim 2. (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in an overall network; col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database)
>
> Civanlar does not explicitly disclose an aggregation tree for routing information. However, Venkatachary discloses wherein comprising the steps:
>
> when the ancestor node of the insertion node is found to exist at or below the maximum aggregation level while determining said presence and absence of the ancestor node, searching for a descendant node of the insertion node; (Venkatachary col. 10, lines 6-33; col 16, ll 26-36: search tree; identify and locate ancestor node)
>
> when a descendant node of the insertion node is found to exist, resetting the aggregation level according to a difference between the prefixes of forwarding information corresponding to the insertion node and the descendant node, and when no descendant nodes of the insertion node are found to exist,

[8] Venkatachary '184's column 15, lines 53-55 reads: "The router searches the source trie of D', and updates the least cost matching filter."

> resetting the aggregation level according to the aggregation Jevel of the ancestor node of the insertion node; (Venkatachary col. 16, lines 26-36: switch pointer; reset and try different branch)
>
> inserting the forwarding information corresponding to the insertion node in the forwarding table when the reset aggregation level is zero; inserting the delegation node representative of the insertion node in the forwarding table when the reset aggregation level is greater than zero; and determining the source area of the inserted routing information, inserting the forwarding information corresponding to the delegation node in the forwarding table when the source area is a virtual area, and inserting the forwarding information corresponding to the insertion node in the forwarding table when the source area is a local area. (Venkatachary col. 15, lines 50-60: search and update (insert) routing information; col. 16, lines 26-36: reset and try different branch)".

Applicant respectfully traverses.

Claim 5 calls for, in part,

> "when the ancestor node of the insertion node is found to exist at or below the maximum aggregation level while determining said presence and absence of the ancestor node, searching for a descendant node of the insertion node;
>
> when a descendant node of the insertion node is found to exist, resetting the aggregation level according to a difference between the prefixes of forwarding information corresponding to the insertion node and the descendant node, and when no descendant nodes of the insertion node are found to exist, resetting the aggregation level according to the aggregation level of the ancestor node of the insertion node;
>
> inserting the forwarding information corresponding to the insertion node in the forwarding table when the reset aggregation level is zero; and
>
> when the reset aggregation level is greater than zero, determining the source area of the inserted routing information, inserting the forwarding information corresponding to the delegation node in the forwarding table when the source area is a virtual area, and inserting the forwarding information

> corresponding to the insertion node in the forwarding table when the source area is a local area."

According to claim 5, in a presence of the ancestor node at or below the maximum aggregation level, a descendant node of the insertion node is searched. When a descendant node of the insertion node is found to exist, the aggregation level is reset according to a difference between the prefixes of forwarding information corresponding to the insertion node and the descendant node; and when no descendant nodes of the insertion node are found to exist, the aggregation level is reset according to the aggregation level of the ancestor node of the insertion node. When the reset aggregation level is zero, the forwarding information corresponding to the insertion node is inserted in the forwarding table. When the reset aggregation level is greater than zero, the source area of the inserted routing information is determined. When the source area is a virtual area, the forwarding information corresponding to the delegation node is inserted in the forwarding table; and when the source area is a local area, the forwarding information corresponding to the insertion node is inserted in the forwarding table.

First, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "when a descendant node of the insertion node is found to exist, resetting the aggregation level according to a difference between the prefixes of forwarding information corresponding to the insertion node and the descendant node", and "when no descendant nodes of the insertion node are found to exist, resetting the aggregation level according to the aggregation level of the ancestor node of the insertion node".

Venkatachary '184's column 16, lines 26-36, as cited by the Examiner, merely teaches how to insert switch pointers in a source trie. Specifically, Venkatachary '184 teaches placing a switch pointer between two nodes **u** and v according to the matching between the nodes and a source prefix.[9] Nowhere in Venkatachary '184 does Venkatachary '184 teach or suggest "reset and try different branch" as asserted by the Examiner.

Secondly, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "inserting the forwarding information corresponding to the delegation node in the forwarding table when the source area is a virtual area, and inserting the forwarding information corresponding to the insertion node in the forwarding table when the source area is a local area". Venkatachary '184's column 15, lines 50-60, as cited by the Examiner, merely mentions that the router may update the least cost matching filter, and it is unclear from Venkatachary '184's specification whether "updating the least cost matching filter" includes "searching and updating routing information" as asserted by the Examiner. Venkatachary '184's column 16, lines 26-36, as cited by the Examiner, merely teaches how to insert switch pointers in a source trie. Nowhere in Venkatachary '184 does Venkatachary '184 teach or suggest "reset and try different branch" as asserted by the Examiner.

In summary, the combination of Civanlar '963 and Venkatachary '184 fails to teach

[9] Venkatachary '184's column 16, lines 44-48 reads: "Let u be a node in T(D) that fails on bit 0; that is, if u corresponds to the source prefix s, then the trie T(D) has no string starting with s0. Let D" be the lowest ancestor of D whose source trie contains a source string starting with prefix s0, say, at node v. Then, the router places a switch pointer at node u pointing to node v."

or suggest the pending claim's "when a descendant node of the insertion node is found to exist, resetting the aggregation level according to a difference between the prefixes of forwarding information corresponding to the insertion node and the descendant node", "when no descendant nodes of the insertion node are found to exist, resetting the aggregation level according to the aggregation level of the ancestor node of the insertion node", and "inserting the forwarding information corresponding to the delegation node in the forwarding table when the source area is a virtual area, and inserting the forwarding information corresponding to the insertion node in the forwarding table when the source area is a local area".

Consequently, the rejection of claim 5 is in error and should be withdrawn.

II-5. Claim 7

Regarding claim 7, on page 10 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 7, Civanlar discloses a method of managing forwarding information. (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in an overall network; col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database)
>
> Civanlar does not explicitly disclose an aggregation tree for routing information. However, Venkatachary discloses wherein comprising the steps:
>
> routing information is deleted from the routing table a deletion node corresponding to the deleted routing information in the aggregation tree; (Venkatachary col. 15, lines 50-60: search and update (insert) forwarding information)
>
> forwarding information corresponding to the deletion node is in a forwarding table, searching for a descendant node

> of the deletion node at a predetermined maximum aggregation level; (Venkatachary col 10, 116-33; col 16, 1126-36: search for a descendent (ancestor) node) and
>
> a descendant node exists for the deletion node at an aggregation level not greater than a predetermined maximum aggregation level, the descendant node as a new source node of a delegation node, and no descendant nodes exist for the deletion node at an aggregation level not greater than a predetermined maximum aggregation level, forwarding information corresponding to the deletion node from the forwarding table. (Venkatachary col. 15, lines 50-60: search and update (delete) forwarding information)".

Applicant respectfully traverses.

Claim 7 calls for, in part,

> "when routing information is to be deleted from a routing table in a distributed router in which all routing nodes share a forwarding information made according to an aggregation tree based on the routing table, detecting a deletion node corresponding to the routing information to be deleted in the aggregation tree;
>
> when forwarding information corresponding to the deletion node is in a forwarding table, searching for a descendant node of the deletion node at a predetermined maximum aggregation level; and
>
> when a descendant node of the deletion node exists at an aggregation level not greater than a predetermined maximum aggregation level, setting the descendant node as a new source node of a delegation node, and when no descendant nodes exist for the deletion node at an aggregation level not greater than a predetermined maximum aggregation level, deleting the forwarding information corresponding to the deletion node from the forwarding table."

According to claim 7, when routing information is to be deleted from a routing table, a position of a deletion node corresponding to the routing information to be deleted is located in the aggregation tree. Then it is determined whether forwarding information corresponding

to the deletion node is in a forwarding table, and when the forwarding information corresponding to the deletion node is in a forwarding table, a descendant node of the deletion node is searched at a predetermined maximum aggregation level. When a descendant node of the deletion node exists at an aggregation level not greater than a predetermined maximum aggregation level, the descendant node is set as a new source node of a delegation node; and when no descendant nodes exist for the deletion node at an aggregation level not greater than a predetermined maximum aggregation level, the forwarding information corresponding to the deletion node is deleted from the forwarding table.

Respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "deleting the forwarding information corresponding to the deletion node from the forwarding table". Venkatachary '184's column 15, lines 50-60, as cited by the Examiner, merely mentions that the router may update the least cost matching filter. It is unclear from Venkatachary '184's specification whether "updating the least cost matching filter" includes "searching and updating (deleting) routing information" as asserted by the Examiner.

Consequently, the rejection of claim 7 is in error and should be withdrawn.

II-6. Claim 9

Regarding claim 9, on page 12 of Paper No. 20080309, the Examiner stated:

> "Regarding Claim 9, Civanlar discloses a distributed architecture router, comprising:
> a switching module accommodating a plurality of routing

> protocols while managing forwarding information within the distributed architecture router; col. 3, lines 41-47: forwarding engine configured to forward new routing table configuration data to every other router port for updating database; col 3, II 37-41: any known types of routing protocols packets may be received (OSPF, RIP, BGP4)) and
>
> a plurality of routing nodes each disposed to service networks within corresponding source areas comprised of local areas, said plurality of routing nodes being connected via said switching module to form a source area comprising a virtual area and share in real time collected routing information assembled by a routing table. (Civanlar col. 2, lines 53-55: coupled to network nodes such as routers/switches in a overall network; col 3, II 37-41: any known types of routing protocols packets may be received (OSPF, RIP, BGP4))
>
> Civanlar does not explicitly disclose an aggregation tree for routing information. However, Venkatachary discloses wherein an aggregation tree derived from routing table information. (Venkatachary col 9, 164 - col 10, I 5: search a tree for a lowest cost match and routing the packet)".

Applicant respectfully traverses.

The pending claim 9 calls for, in part,

> "a switching module accommodating a plurality of routing protocol processing units while managing forwarding information within the distributed architecture router, with each routing protocol processing unit processing data in accordance with a respectively corresponding routing protocol; and
>
> a plurality of routing nodes each disposed to service networks within corresponding source areas comprised of local areas , with each routing node having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units in switching module to form a source area comprising a virtual area and share in real time collected routing information assembled by a routing table and an aggregation tree derived from said routing table."

According to the pending claim 9 in connection with Applicant's original FIG. 3, a

distributed router includes a switching module 250 and a plurality of routing nodes 210. Switching module 250 includes a plurality of routing protocol processing units 251-254 each processing data in accordance with a respective corresponding routing protocol (i.e., RIPD, SDPFD, BGPD, or DGLUED). Each one of a plurality of routing nodes 210 has a plurality of routing protocol processing units 211-214 communicatively connected with routing protocol processing units 251-254 in switching module 250.

First, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "plurality of routing protocol processing units" and "each routing protocol processing unit processing data in accordance with a respective corresponding routing protocol". Civanlar '963's column 3, lines 37-41 as cited by the Examiner in Paper No. 20080309 merely mentions different types of routing protocols packets.[10] Civanlar '963 does not disclose however, how to process those different types of routing protocols packets, or whether those different types of routing protocols packets are processed in the same processing unit or in different processing units. On the other hand, Applicant's claim 1 defines that each routing protocol processing unit processes data in accordance with a respectively corresponding routing protocol (i.e., RIPD, SDPFD, BGPD, or DGLUED). In other words, in the pending claim 1's distributed router, data with different protocol is processed in a corresponding routing protocol processing unit.

---

[10] Civanlar '963's column 3, lines 37-41 reads: "Any known types of routing protocols packets may be received by the routing engine 107, such as those conforming to the routing Internet protocol (RIP), the open shortest path forwarding (OSPF) protocol, or the border gateway protocol 4 (BGP4)."

Secondly, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "each routing node having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units in switching module". Civanlar '963 merely mentions that the switching fabric may partially or fully interconnect some or all of the router ports.[11] Civanlar '963 does not disclose however, whether there are a plurality of routing protocol processing units in the switching fabric, and how to interconnect the switching fabric with the router ports. On the other hand, Applicant's pending claim 1 clearly defines that the routing protocol processing units in the switching module are connected with the corresponding routing protocol processing units in each of the routing nodes.

In summary, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "plurality of routing protocol processing units", or "each routing protocol processing unit processing data in accordance with a respective corresponding routing protocol", or "each routing node having a plurality of routing protocol processing units communicatively connected with the corresponding routing protocol processing units in switching module".

As a result, the rejection of claim 9 should be withdrawn.

II-7. Claim 10

---

[11] Civanlar '963's column 3, lines 5-7 reads: "The switching fabric 102 may switch between and/or otherwise partially or fully interconnect some or all of the intelligent router ports 103."

Regarding claim 10, on pages 13-14 of Paper No. 20080309, the Examiner stated:

> "Civanlar does not explicitly disclose an aggregation tree for routing information.
>
> However, Venkatachary discloses wherein:
>
> identifying in said aggregation tree a position for addition of an insertion node corresponding to said new routing information; (Venkatachary col. 15, lines 50-60: search and update (insert) forwarding information)
>
> making a search of said aggregation tree within a maximum aggregation level to identify an ancestor node of said insertion node; (Venkatachary col. 10, lines 6-33; col 16, 1126-36: search tree; identify and locate ancestor node)
>
> forgoing updating of said forwarding table with forwarding information
> corresponding to said insertion node when said insertion node and said ancestor node were generated the same said source area and said search identifies said ancestor node; (Venkatachary col. 16, lines 37-52: no update; no node at location)
>
> resetting said maximum aggregation leve[to a reset aggregation level not less than said maximum aggregation level when said search fails to identify said ancestor node and adding a delegation node representative of said insertion node at said reset aggregation level; (Venkatachary col. 16, lines 26-36: switch pointer; reset and try different branch)
>
> making an identification of said source area of said new routing information; inserting said forwarding information corresponding to said delegation node when said identification establishes that said source area of said new routing information is a virtual area; (Venkatachary col. 15, lines 50-60: search and update (insert) forwarding information) and
>
> inserting said forwarding information corresponding to said delegation node when said identification establishes that said source area of said new routing information is a [ocal area. (Venkatachary col. 15, lines 50-60: search and update (insert) forwarding information)".

Applicant respectfully traverses.

Claim 10 calls for, in part,

> "identifying in said aggregation tree a position for addition of an insertion node corresponding to said new routing information;
>
> making a search of said aggregation tree within a maximum aggregation level to identify an ancestor node of said insertion node;
>
> forgoing updating of said forwarding table with forwarding information corresponding to said insertion node when said insertion node and said ancestor node were generated from the same source area and said search identifies said ancestor node;
>
> resetting said maximum aggregation level to a reset aggregation level not less than said maximum aggregation level when said search fails to identify said ancestor node and adding a delegation node representative of said insertion node at said reset aggregation level;
>
> making an identification of said source area of said new routing information;
>
> inserting said forwarding information corresponding to said delegation node when said identification establishes that said source area of said new routing information is a virtual area; and
>
> inserting said forwarding information corresponding to said delegation node when said identification establishes that said source area of said new routing information is a local area."

According to claim 10, when a piece of new routing information is to be inserted into a routing table, first a position of an insertion node corresponding to the new routing information is detected. It is then determined whether there is an ancestor node of the insertion node at or below a predetermined maximum aggregation level with respect to the insertion node. If there is an ancestor node, the forwarding table is not updated with the information corresponding to the insertion node when both of the insertion node and the ancestor node have been generated from a common source area. If there is no ancestor node, the aggregation level is reset and a delegation node representative of the insertion node is

inserted at the reset aggregation level. When the source area of the new routing information is a virtual area, the forwarding information corresponding to the delegation node is inserted in the forwarding table, and when the source area of the new routing information is a local area, the forwarding information corresponding to the insertion node is inserted in the forwarding table.

First respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "inserting routing information in a routing table". In Paper No. 20080309, the Examiner explicitly admitted that Civanlar '963 fails to disclose an aggregation tree for routing information. On the other hand, Venkatachary '184 merely teaches how to place a switch pointer between two nodes in a prefix tree. In point of fact, Venkatachary '184's "switch pointer" is different from the pending claim's "node". As explained in Venkatachary '184's specification, the switch pointer is pointed from a first node to a second node to enable a direct "jump" from the first node to the second node when the search on the first node fails.[12] There is no teaching in Venkatachary '184 regarding how to insert a piece of new routing information.

Secondly, respectfully, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "when the source area of the new routing information is a virtual area, the forwarding information corresponding to the delegation node is inserted in the forwarding table, and when the source area of the new routing information is a local

[12] Venkatachary '184's column 16, lines 31-33 reads: "Intuitively, the switch pointers allow a jump directly to the lowest point in the ancestor source trie that has at least as good a source match as the current node."

area, the forwarding information corresponding to the insertion node is inserted in the forwarding table". Venkatachary '184's column 15, lines 50-60 as cited by the Examiner merely mentions that the router may update the least cost matching filter.[13] A thorough search of Venkatachary '184's specification demonstrates that Venkatachary '184 does not disclose how to "update the least cost matching filter". Venkatachary '184 does not however, teaches inserting new routing information, or determining whether the source area of the routing information is a virtual area or a local area.

In summary, the combination of Civanlar '963 and Venkatachary '184 fails to teach or suggest the pending claim's "inserting routing information in a routing table", or "when the source area of the new routing information is a virtual area, the forwarding information corresponding to the delegation node is inserted in the forwarding table, and when the source area of the new routing information is a local area, the forwarding information corresponding to the insertion node is inserted in the forwarding table".

Consequently, the rejection of claim 10 is in error and should be withdrawn.

In view of the foregoing amendments and remarks, all claims are deemed to be allowable and this application is believed to be in condition to be passed to issue. If there are any questions, the examiner is asked to contact the applicant's attorney.

[13] Venkatachary '184's column 15, lines 53-55 reads: "The router searches the source trie of D', and updates the least cost matching filter."

No fee is incurred by this Amendment.

Respectfully submitted,

Robert E. Bushnell
Attorney for the Applicant
Registration No.: 27,774

2029 "K" Street N.W., Suite 600
Washington, D.C. 20006-1004
(202) 408-9040

Folio: P57009
Date: 1/26/09
I.D.: REB/YFM